WRDC-TR-90-8007
Volume V
Part 33

INTEGRATED INFORMATION SUPPORT SYSTEM (IISS)
Volume V - Common Data Model Subsystem
Part 33 - Define/Construct the Neutral Data Definition Language (NDDL) for the CDM Subsystem User Manual

M. Apicella, S. Singh

Control Data Corporation
Integration Technology Services
2970 Presidential Drive
Fairborn, OH 45324-6209

September 1990

Final Report for Period 1 April 1987 - 31 December 1990

MANUFACTURING TECHNOLOGY DIRECTORATE
WRIGHT RESEARCH AND DEVELOPMENT CENTER
AIR FORCE SYSTEMS COMMAND
WRIGHT-PATTERSON AIR FORCE BASE, OHIO 45433-6533

NOTICE

When Government drawings, specifications, or other data are used for any purpose other than in connection with a definitely related Government procurement operation, the United States Government thereby incurs no responsibility nor any obligation whatsoever, regardless whether or not the government may have formulated, furnished, or in any way supplied the said drawings, specifications, or other data. It should not, therefore, be construed or implied by any person, persons, or organization that the Government is licensing or conveying any rights or permission to manufacture, use, or market any patented invention that may in any way be related thereto.

This technical report has been reviewed and is approved for publication.

This report is releasable to the National Technical Information Service (NTIS). At NTIS, it will be available to the general public, including foreign nations

DAVID L. JUDSON, Project Manager
WRDC/MTI
Wright-Patterson AFB, OH 45433-6533

25 July 91
DATE

FOR THE COMMANDER:

BRUCE A. RASMUSSEN, Chief
WRDC/MTI
Wright-Patterson AFB, OH 45433-6533

25 July 91
DATE

If your address has changed, if you wish to be removed form our mailing list, or if the addressee is no longer employed by your organization please notify WRDC/MTI, Wright-Patterson Air Force Base, OH 45433-6533 to help us maintain a current mailing list.

Copies of this report should not be returned unless return is required by security considerations, contractual obligations, or notice on a specific document.

Unclassified

SECURITY CLASSIFICATION OF THIS PAGE

REPORT DOCUMENTATION PAGE

1a. REPORT SECURITY CLASSIFICATION Unclassified	1b. RESTRICTIVE MARKINGS
2a. SECURITY CLASSIFICATION AUTHORITY	3. DISTRIBUTION/AVAILABILITY OF REPORT Approved for Public Release; Distribution is Unlimited.
2b. DECLASSIFICATION/DOWNGRADING SCHEDULE	
4. PERFORMING ORGANIZATION REPORT NUMBER(S) UM 620341401	5. MONITORING ORGANIZATION REPORT NUMBER(S) WRDC-TR- 90-8007 Vol. V, Part 33
6a. NAME OF PERFORMING ORGANIZATION Control Data Corporation; Integration Technology Services 6b. OFFICE SYMBOL *(if applicable)*	7a. NAME OF MONITORING ORGANIZATION WRDC/MTI
6c. ADDRESS *(City, State, and ZIP Code)* 2970 Presidential Drive Fairborn, OH 45324-6209	7b. ADDRESS *(City, State, and ZIP Code)* WPAFB, OH 45433-6533
8a. NAME OF FUNDING/SPONSORING ORGANIZATION Wright Research and Development Center, Air Force Systems Command, USAF 8b. OFFICE SYMBOL *(if applicable)* WRDC/MTI	9. PROCUREMENT INSTRUMENT IDENTIFICATION NUM. F33600-87-C-0464
8c. ADDRESS *(City, State, and ZIP Code)* Wright-Patterson AFB, Ohio 45433-6533	10. SOURCE OF FUNDING NOS.

PROGRAM ELEMENT NO.	PROJECT NO.	TASK NO.	WORK UNIT NO.
78011F	595600	F95600	20950607

11. TITLE *(Include Security Classification)* [See block 19	
12. PERSONAL AUTHOR(S) Control Data Corporation: Apicella, M. L., Singh, S.	

13a. TYPE OF REPORT	13b. TIME COVERED	14. DATE OF REPORT *(Yr.,Mo.,Day)*	15. PAGE COUNT
Final Report	4/1/87-12/30/90	1990 September 30	29

16. SUPPLEMENTARY NOTATION

WRDC/MTI Project Priority 6203

17. COSATI CODES			18. SUBJECT TERMS *(Continue on reverse if necessary and identify block no.)*
FIELD	GROUP	SUB GR.	
1308	0905		

19. ABSTRACT *(Continue on reverse if necessary and identify block number)*

This document defines the requirements for constructing the Neutral Data Definition Language (NDDL) which consists of the precompilation of 393 Neutral Data Manipulation Language (NDML) routines into a single logical unit of work. Included in this document are lists of 25 groups containing NDML routines. Also included are examples of a precompilation of one of these 25 groups, and an example of an insertion into the NDDL Request Processor Main Program's object library.

BLOCK 11:

INTEGRATED INFORMATION SUPPORT SYSTEM
Vol V - Common Data Model Subsystem

Part 33 - Define/Construct the Neutral Data Definition Language (NDDL) for the CDM Subsystem User Manual

20. DISTRIBUTION/AVAILABILITY OF ABSTRACT	21. ABSTRACT SECURITY CLASSIFICATION	
UNCLASSIFIED/UNLIMITED x SAME AS RPT. DTIC USERS	Unclassified	
22a. NAME OF RESPONSIBLE INDIVIDUAL David L. Judson	22b. TELEPHONE NO. *(Include Area Code)* (513) 255-7371	22c. OFFICE SYMBOL WRDC/MTI

EDITION OF 1 JAN 73 IS OBSOLETE

DD FORM 1473, 83 APR

Unclassified

SECURITY CLASSIFICATION OF THIS PAGE

FOREWORD

This technical report covers work performed under Air Force Contract F33600-87-C-0464, DAPro Project. This contract is sponsored by the Manufacturing Technology Directorate, Air Force Systems Command, Wright-Patterson Air Force Base, Ohio. It was administered under the technical direction of Mr. Bruce A. Rasmussen, Branch Chief, Integration Technology Division, Manufacturing Technology Directorate, through Mr. David L. Judson, Project Manager. The Prime Contractor was Integration Technology Services, Software Programs Division, of the Control Data Corporation, Dayton, Ohio, under the direction of Mr. W. A. Osborne. The DAPro Project Manager for Control Data Corporation was Mr. Jimmy P. Maxwell.

The DAPro project was created to continue the development, test, and demonstration of the Integrated Information Support System (IISS). The IISS technology work comprises enhancements to IISS software and the establishment and operation of IISS test bed hardware and communications for developers and users.

The following list names the Control Data Corporation subcontractors and their contributing activities:

SUBCONTRACTOR	ROLE
Control Data Corporation	Responsible for the overall Common Data Model design development and implementation, IISS integration and test, and technology transfer of IISS.
D. Appleton Company	Responsible for providing software information services for the Common Data Model and IDEF1X integration methodology.
ONTEK	Responsible for defining and testing a representative integrated system base in Artificial Intelligence techniques to establish fitness for use.
Simpact Corporation	Responsible for Communication development.

Structural Dynamics Research Corporation	Responsible for User Interfaces, Virtual Terminal Interface,and Network Transaction Manager design, development, implementation, and support.
Arizona State University	Responsible for test bed operations and support.

TABLE OF CONTENTS

Page

SECTION 1. Introduction 1-1

SECTION 2 Define NDML Groups 2-1

SECTION 3. Precompiling Process for NDDL 3-1

SECTION 4. Procedures to Construct NDDL 4-1

SECTION 1

Introduction

The construction of NDDL requires the results of the precompilation of 438 NDML routines into a single logical unit of work. These NDML routines have been grouped into 12 units, each of which must be precompiled. After completion of precompilation for all the modules, the following steps must be executed in order to construct the NDDL executable:

- Generate the NDDL Request Processor Main Program
- Compile and insert into the object library (GENLIB) the generated NDDL Request Processor Main Program
- Create the NDDL Executables

Section II of this document lists the 12 groups and each NDML routine contained in the group. Section III contains an explanation of the process used to precompile the 12 groups. Section IV contains an example of generating, compiling and insertion into the object library of the NDDL Request Processor Main Program.

SECTION 2

Define NDML Groups

This section defines the 12 groups to be precompiled that are required for NDDL, the application name associated with each group and the NDML routines contained in each group. Each group has been returned to CM as an NDDLn.TST file (eg: NDDL12.TST), where n is the group number.

Group 1

Application Name - NDDL1

ADDKWA
ADDKWE
ADDKWR
ADDMIG
ALLDB
ALLDBMS
ALLDOM
ALLDT
ALLECRT
ALLMENT
ALLMREC
ALLREC
ALLRNG
ALLSET
ALLVALU
ALLVIEW
ALT1ERT
ALTAISM
ALTDBT
BLKCL1
BLKCLST
CHKATT
CHKAUCV
CHKHPRT
CHKREL
CHKSTMP
CMBACAL
CMBALI
CMBEKW
CMBOA
CMBRKW
COPAALI
COPAKW
COPEKW
COPOA
COPRKW
COPVALI
COPYAC
CPY1SMD
DEL1AIM
DEL1ASM
DEL1ERT
DEL1PDF

DEL1PRF
DELAAA
DELACON

Group 2

Application Name - NDDL2

DELAISM
DELAPRM
DELAREA
DELASM
DELCONI
DELDBDF
DELDBH
DELDBMS
DELDBRT
DELDBST
DELDIV
DELDOM
DELDSCT
DELDT
DELDTD
DELDTNO
DELECON
DELECRF
DELHPN
DELHPNA
DELHST
DELMDKC
DELMDRC
DELOAC
DELPARM
DELPDF
DELPDI
DELPSB
DELQCRF

Group 3

Application Name - NDDL3

DELSMOD
DELUNIN
DELVIEW
DEPFROM
DEPKCM
DLMDAUC
DOMUSAG
DRPAC
DRPALTG
DRPDFMP
DRPDIV
DRPECRT
DRPMGKM
DRPMGRC
DRPRULE
DRPSTMP
DRPVAL
DRPVALA
ECRTALL
ECRTCHK
FCOPATT
FILEINS
FINDDOM
FND1MEM
FNDACM
FNDASA
FNDASM
FNDRCM
GEN1DSC
GENAKW
GENALG
GENCMPX
GENCREC
GENCRPT
GENCRUN
GENDEP
GENDESC
GENECRT
GENEHP
GENOA
GENPSB
GENRKW
GENTODF
GENTOST
GETACAL

Group 4

Application Name - NDDL4

DELQCTG
GTDIPR
GTDTTYP
GTVIEWS
INDFROM
INS1PSB
INSAISM
INSAPRM
INSAREA
INSAUCS
INSCOPA
INSDAA
INSDB
INSDI
INSDIPA
INSDOM
INSDSCT
INSDT
INSECRF
INSECRT
INSHST
INSPARM
INSPART

Group 5

Application Name - NDDL5

INSPCB
INSQCTG
INSRNG
INSRTPA
INSRTYP
INSRULE
INSSCH
INSSMOD
INSTGPA
INSUNIN
INSVAL
INSVIEW
MODATT
MODDBAL
MODENT
MODKC
MODLOC
MODOATT
MODPCB
MODPWRD
MODSCH
NCOMMIT
NROLBAK
OLDTAGM
RETACKW
RETATTR
RETDT
RETECKW
RETRAC1
RETRAC2
RETRACP
RETRCKW
RETREC1
RETREC2
RETRECP
RETRNGA

Group 6

Application Name - NDDL6

SEL1DSC
SELACNM
SELAIMP
SELAIMT
SELAISM
SELAPRM
SELDB
SELDBAA
SELDBNM
SELDESC
SELDFPM
SELDI
SELDSTP
SELECNM
SELECXR
SELHOST
SELRMAP
SELHP
SELHSTS
SELIKEY
SELKCM
SELKCMA
SELMODS
SELMREC
SELMTAG
SELPARM
SELPEC
SELRCNM
SELRELC
SELRSET
SELRTPM
SELSTM
SELTGEC
SELUNIN
SELURT
STRINS
UPD1PSB
UPDAUCS
UPDIND
UPDRULE
UPDSMOD
UPDTDT
VALVWRC
VERACDT
VERACNM
VERAIM
VERAIMD
VERAIMR
VERAISM

Group 7

Application Name - NDDL7

VERALG
VERALGI
VERALI
VERALMP
VERAPDF
VERAPRM
VERAREA
VERARL
VERATT
VERATTP
VERAUC
VERCRC
VERDB
VERDBA
VERDBAA
VERDBAL
VERDBAS
VERDBH
VERDBMS
VERDBRL
VERDBRT
VERDBST
VERDF
VERDFID
VERDFDT
VERDFLD
VERDFPA
VERDFUS
VERDI
VERDIDN
VERDIDT
VERDIID
VERDIPA
VERDOM
VERDPSB
VERDSCT
VERDSL3
VERDSTP
VERDT
VERDTD
VERDTFL
VERDTN
VERDTYP
VERDUNI
VERECN

Group 8

Application Name - NDDL8

VERECR
VERECRT
VERENT
VERENTP
VERERMD
VERHORZ
VERHPDF
VERHPN
VERHPST
VERHST
VERHZP
VERITAG
VERKC
VERKCM
VERKCMG
VERKW
VERKWE
VERKWR
VERLUW
VERLWMD
VERMAPD
VERMOD
VERMPDF
VERMPDT
VERMUNI
VERNMA
VERNME
VEROAC
VERPARM
VERPART
VERPASS
VERPCB
VERPSB
VERRCB
VERRCBS
VERRCC
VERRCMP
VERRCST
VERREC
VERRNG
VERRPS
VERRSET
VERRSUS

Group 9

Application Name - NDDL9

DELLUW
VERRT
VERRTID
VERRTNO
VERRULE
VERSCH
VERSDT
VERSMOD
VERSMS
VERSTID
VERTAG
VERTAP
VERTAUC
VERTPDI
VERTYP
VERUDT
VERUDTN
VERUNIN
VERVAL
VERVIEW
VERVWQU
VOMAPS
WRTACKW
WRTALI
WRTANAM
WRTDITM
WRTECKW
WRTENAM
WRTSLCT
WRTWHCL
DRPPRF1
DRPRCE
DRPRNG
DRPRNGA
FNDAUC
FNDECM
FNDOAC
FNDPDF
GENEKW
GENEUN
GENIND
GENKEY

Group 10

Application Name - NDDL10

GETMEMB
GETRCID
GTAUCPR
GTCNTPR
INSDBH
INSDBMS
INSDESC
INSDFPA
INSPDF
INSPDI
INSPWRD
INSQCRF
KEYLOOK
MGENOA
MODACEC
MODAISM
OUTDESC
RACKW2
RECKW2
RELKW
RETSTD
RETVALA
RRCKW2

Group 11

Application Name - NDDL11

ADDMIGC
ALLMROJ
ALMROJC
AOACREC
AOACTAG
CATFROM
CHKCAT
CHKCDOM
CHKMKM
CHKRELC
CMCOUNT
CMPCRDA
COPCKW
DELROJ
DELROJE
DEPKCM1
DRPCRE
DRPCRE1
FNDECCM
FNDECCR
FNDECM1
GENCAT
GENCNM1
GENCNM2
GENCNM3
GENCNME
GENFROM
GENGENC
GETCRID

Group 12

Application Name - NDDL12

INS1ROJ
INSKC
SELCATC
SELCATM
SELCRAD
SELCRDN
SELDBMS
SELDBRT
SELRELL
VALROJ
VERAODM
VERCMSV
VERCMU
VERCR
VERCRDA
VERCRTC
VERKCT
VERKCT1
VERLR
VERRCT
VERROJ
VERROJR
GETCARD
GETDFSL
GETDOM
GETDTN
GETECAL
GETECS

SECTION 3

Precompiling Process for NDDL

PRENDDL.COM is a procedure that submits one batch job for each group of files to be precompiled. Each batch job will precompile the routines and compile the resulting generated code. The procedure that is actually submitted is BLDNDDL.COM. All 12 batch jobs are submitted so that they are synchronized to start one job after the previous one is finished. To begin the process, type:

```
$ @CDMDIR:[COM]PRENDDL.COM  EVE_BATCH
```

SECTION 4

Procedures to Construct NDDL

After successful completion of the precompilation of all NDML group defined in Section II, the following steps must be executed to construct the NDDL executable.

1. Execute the procedure file GENRPD.COM to generate the main request processor for NDDL. Proceed as follows:

   ```
   $ @GENRPD

   GENERATE REQUEST PROCESSOR DRIVERS
   ----------------------------------

   ENTER LOG UNIT WORK NAME                        NDDL
   ENTER ORACLE USERNAME/PASSWORD (CDM/CDM)        CDM/CDM
   >>> GENERATION FINISHED <<<
   RESULTS OF GENERATION CAN BE
   FOUND ON FILE NDDL.RPD
   ```

   Review results of generation in file NDDL.RPD. On examining NDDL.RPD, note the module name (xxxxx) and the file name (yyyyy.TMP).

2. Compile and insert into the object library (GENLIB) the generated request processor main. Proceed as follows:

   ```
   $ @MCMPDDL    yyyyy.TMP
   ```

3. Create the executable file for the batch version of NDDL. Proceed as follows:

   ```
   $ @LNKNDDL
   ```

4. Create the executable file for the UIMS version of NDDL. Proceed as follows:

   ```
   $ @ULNKDDL
   ```

   The following pages contain listings for:

   PRENDDL.COM
   BLDNDDL.COM
   GENRPD.COM
   MCMPDDL.COM
   LNKNDDL.COM
   ULNKDDL.COM

```
$!
$!
$!
$!      PRENDDL.COM -  proc used to precompile in batch mode.
$!                     this will submit each group after the
```

```
$!                      previous group has completed.  A log
$!                      file will be produced for each group
$!                      so that you may check the status of
$!                      the job.  this proc MUST be submitted
$!                      in BATCH mode as well since the SYNCHRONIZE
$!                      command will lock the terminal until the
$!                      previous job is complete.
$!
$!                      Each group of PRECOMPILES will produce
$!                      many output files that are not needed
$!                      unless an error is encountered during the
$!                      build.  These include NDDLx.IN, NDDLx.DAT,
$!                      NDDLx.MSG, NDDLx.ERR, as well as the listings
$!                      produced during the compilation of the .TMP
$!                      files.  All of these files should be delete
$!                      as you go to ensure there is enough disk
                          space.
$!
$!      PARAMETERS:     P1 = The batch que were the jobs will
                          execute.
$!                      If P1 is not specified processing will
                          terminate
$!                      and an error message will be issued.
$!
$ IF P1 .EQS. "" then goto nobatch
$SUBMIT/NOTIFY/KEEP/NOPRINTER/NAME=NDDL1/QUE='P1'/-
       PARAMETER=CDMDIR:[NDDL]NDDL1 CDMDIR:[COM]BLDNDDL.COM
$SYNCHRONIZE/QUE='P1' NDDL1
$SUBMIT/NOTIFY/KEEP/NOPRINTER/NAME=NDDL2/QUE='P1'/-
       PARAMETER=CDMDIR:[NDDL]NDDL2 CDMDIR:[COM]BLDNDDL.COM
$SYNCHRONIZE/QUE='P1' NDDL2
$SUBMIT/NOTIFY/KEEP/NOPRINTER/NAME=NDDL3/QUE='P1'/-
       PARAMETER=CDMDIR:[NDDL]NDDL3 CDMDIR:[COM]BLDNDDL.COM
$SYNCHRONIZE/QUE='P1' NDDL3
$SUBMIT/NOTIFY/KEEP/NOPRINTER/NAME=NDDL4/QUE='P1'/-
       PARAMETER=CDMDIR:[NDDL]NDDL4 CDMDIR:[COM]BLDNDDL.COM
$SYNCHRONIZE/QUE='P1' NDDL4
$SUBMIT/NOTIFY/KEEP/NOPRINTER/NAME=NDDL5/QUE='P1'/-
       PARAMETER=CDMDIR:[NDDL]NDDL5 CDMDIR:[COM]BLDNDDL.COM
$SYNCHRONIZE/QUE='P1' NDDL5
$SUBMIT/NOTIFY/KEEP/NOPRINTER/NAME=NDDL6/QUE='P1'/-
       PARAMETER=CDMDIR:[NDDL]NDDL6 CDMDIR:[COM]BLDNDDL.COM
$SYNCHRONIZE/QUE='P1' NDDL6
$SUBMIT/NOTIFY/KEEP/NOPRINTER/NAME=NDDL7/QUE='P1'/-
       PARAMETER=CDMDIR:[NDDL]NDDL7 CDMDIR:[COM]BLDNDDL.COM
$SYNCHRONIZE/QUE='P1' NDDL7
$SUBMIT/NOTIFY/KEEP/NOPRINTER/NAME=NDDL8/QUE='P1'/-
       PARAMETER=CDMDIR:[NDDL]NDDL8 CDMDIR:[COM]BLDNDDL.COM
$SYNCHRONIZE/QUE='P1' NDDL8
$SUBMIT/NOTIFY/KEEP/NOPRINTER/NAME=NDDL9/QUE='P1'/-
       PARAMETER=CDMDIR:[NDDL]NDDL9 CDMDIR:[COM]BLDNDDL.COM
$SYNCHRONIZE/QUE='P1' NDDL9
$SUBMIT/NOTIFY/KEEP/NOPRINTER/NAME=NDDL10/QUE='P1'/-
       PARAMETER=CDMDIR:[NDDL]NDDL10 CDMDIR:[COM]BLDNDDL.COM
$SYNCHRONIZE/QUE='P1' NDDL10
$SUBMIT/NOTIFY/KEEP/NOPRINTER/NAME=NDDL11/QUE='P1'/-
       PARAMETER=CDMDIR:[NDDL]NDDL11 CDMDIR:[COM]BLDNDDL.COM
$SYNCHRONIZE/QUE='P1' NDDL11
$SUBMIT/NOTIFY/KEEP/NOPRINTER/NAME=NDDL12/QUE='P1'/-
```

```
       PARAMETER=CDMDIR:[NDDL]NDDL12 CDMDIR:[COM]BLDNDDL.COM
$SYNCHRONIZE/QUE='P1' NDDL12
$exit
$NOBATCH:
$    WRITE SYS$OUTPUT "A batch que must be entered as a parameter."
$    WRITE SYS$OUTPUT "Please resubmit the job with the proper Que."
$exit
```

```
$!
$!     BLDNDDL.COM
$!
$!     RECEIVE A TEST FILE OF PRC'S AND PRECOMPILE AND COMPILE THEM
$!
$!        P1 = THE NAME OF THE FILE CONTAIN THE PRC's TO BE
                 PRECOMPILED
$!             IN THE FORM:  CDMDIR:[NDDL]NDDL1
$!
$!
$!
$WS:= WRITE SYS$OUTPUT
$DEFINE IISSGLIB "CDMDIR:[NDDL.TEMPS]GENOLB.OLB"
$DEFINE CDMTEMPS CDMDIR:[NDDL.TEMPS]
$ IF P1 .NES. "" THEN GOTO PARMGO
$WS "PRECOMPILE AND COMPILE A GROUP OF APPLICATION PROGRAMS"
$WS "-------------------------------------------------"
$! read an input file containing names of modules to be precompiled
$!
$INQUIRE AP " NAME OF THE APPLICATION>"
$ GOTO NOPARM
$PARMGO:
$ AP = P1
$NOPARM:
$ CREATE 'AP'.DAT
$ OPEN/WRITE   NDMLIN  'AP'.IN
$ OPEN/READ NDDLIN 'AP'.TST
$NEXT:
$ READ/END_OF_FILE=INDONE NDDLIN FILE
$ WS "FILE:''File'"
$ APPEND CDMDIR:[NDDL]'FILE'.PRC 'AP'.DAT
$ GOTO NEXT
$INDONE:
$ CLOSE NDDLIN
$!
$!  5/24/88:  FDL stuff added in because GENRPD now requires fixed
$!            length .DAT files since conversion to FIOPS.
$!
$ OPEN/WRITE FDLIN CDMDIR:[COM]FIX.FDL
$ WRITE FDLIN "IDENT   ""23-FEB-1988 09:49:43   VAX-11 FDL
Editor"""
$ WRITE FDLIN "    "
$ WRITE FDLIN "SYSTEM"
$ WRITE FDLIN "        SOURCE                  VAX/VMS"
$ WRITE FDLIN "    "
$ WRITE FDLIN "FILE"
$ WRITE FDLIN "        ALLOCATION              391"
$ WRITE FDLIN "        BEST_TRY_CONTIGUOUS     yes"
$ WRITE FDLIN "        EXTENSION               39"
$ WRITE FDLIN "        ORGANIZATION            sequential"
$ WRITE FDLIN "    "
$ WRITE FDLIN "RECORD"
$ WRITE FDLIN "        BLOCK_SPAN              yes"
$ WRITE FDLIN "        CARRIAGE_CONTROL        carriage_return"
$ WRITE FDLIN "        FORMAT                  fixed"
$ WRITE FDLIN "        SIZE                    80"
$ CLOSE FDLIN
$ CONVERT/PAD=%O40/FDL=CDMDIR:[COM]FIX.FDL 'AP'.DAT 'AP'.DAT
$ WRITE NDMLIN "NDDL VAX VAX"," COBOL NDML C ",AP,".DAT ",AP,".ERR
```

```
","CDM/CDM"," FD=N"
$ OPEN/WRITE EDIPRC CDMDIR:[COM]EDIT.PROC
$ WRITE EDIPRC "S/80/200/WH"
$ WRITE EDIPRC "EXIT"
$ CLOSE EDIPRC
$ CLOSE NDMLIN
$ EDIT/COMMAND=CDMDIR:[COM]EDIT.PROC CDMDIR:[COM]FIX.FDL
$ DEASSIGN SYS$OUTPUT
$ CONVERT/PAD=%040/FDL=CDMDIR:[COM]FIX.FDL 'AP'.IN 'AP'.IN
$ DELETE CDMDIR:[COM]FIX.FDL;*, CDMDIR:[COM]EDIT.PROC;*
$!
$!  INPUTS TO PRECOMPILER ARE NOW SET UP
$!   GO AHEAD AND RUN IT:
$!
$  ASSIGN/USER_MODE SYS$COMMAND SYS$INPUT
$! ASSIGN  'AP'.IN NDML
$ RENAME 'AP'.IN CDMDIR:[NDDL]NDML.DAT
$ ASSIGN 'AP'.OUT SYS$OUTPUT
$ SET DEFAULT CDMDIR:[NDDL]
$ RUN CMDIR:[RUNAREA]NDML.EXE
$ RENAME CDMDIR:[NDDL]NDML.DAT 'AP'.IN
$!
$ ALLDONE:
$ DEASSIGN SYS$OUTPUT
$!
$!  check the .out file for errors in precompiling
$!
$OPEN/READ EFLE 'AP'.OUT
$ZR:="0"
$  NERRLOOP:
$   READ/END_OF_FILE=COMPERR EFLE EREC
$     LENG = 'F$LENGTH(EREC)'
$     UN = 'F$LOCATE("UNSUCC",EREC)'
$     IF 'UN' .EQS. 'LENG' THEN GOTO NERRLOOP
$     UN1 = 'UN' - 13
$     UN2 = 'F$EXTRACT(UN1,1,EREC)'
$     IF UN2 .EQS. ZR THEN GOTO NDMLGOOD
$WS "THE PRECOMPILE OF ''AP' HAS ''UN2'  UNSUCCESSFUL ROUTINES"
$WS "CHECK THE ''AP'.ERR FILE FOR ERRORS"
$GOTO EXIT
$COMPERR:
$WS "PRECOMPILE FAILED"
$GOTO EXIT
$!
$!  the precompile was successful, compile the code
$!
$  NDMLGOOD:
$WS "    "
$WS "NDML PRECOMPILE SUCCESSFULLY COMPLETED"
$WS "BEGIN COMPILING GENERATED CODE"
$  NDMLGDRD:
$ READ/END_OF_FILE = COMPERR EFLE EREC
$  LENG = 'F$LENGTH(EREC)'
$  UN = 'F$LOCATE("==> USE", EREC)'
$  IF 'UN' .EQS. 'LENG' THEN GOTO NDMLGDRD
$!
$ASSIGN 'AP'.MSG SYS$OUTPUT
$UN1 = 'UN' + 8
$NNAM:='F$EXTRACT(UN1,30,EREC)'
```

```
$WS NNAM
$CLOSE EFLE
$ @'NNAM'
$DEASSIGN SYS$OUTPUT
$purge 'AP'.*
$WS "RESULTS OF COMPILE CAN BE FOUND ON ''AP'.MSG"
$EXIT:
$DEFINE IISSGLIB "CDMDIR:[TEST]GENOLB.OLB"
$deassign cdmtemps
```

```
$!
$!
$!    GENRPDDBG.COM
$!    THIS PROC WILL CREATE GENRPD.DAT FOR RUNNING GENRPD . GENRPD
$!    GENERATES AN RP-MAIN PROGRAM. PROMPTS FOR LOGICAL UNIT OF WORK
$!    NAME AND ORACLE USER NAME/PASSWORD
$!    OUTPUT FOUND ON 'LUW'.RPD
$!
$!    MODIFIED: JAN 10, 1987
$!
$ WRITE SYS$OUTPUT "    "
$ WRITE SYS$OUTPUT " GENERATE REQUEST PROCESSOR DRIVERS"
$ WRITE SYS$OUTPUT " ---------------------------------"
$ INQUIRE LUW "ENTER LOG UNIT WORK NAME "
$ INQUIRE UNPW "ENTER ORACLE USERNAME/PASSWORD(CDM/CDM) "
$ MYHOST = "VAX"
$ CREATE GENRPD.DAT
$ OPEN/WRITE GENRPD.DAT GENRPD.DAT
$ WRITE GENRPD.DAT LUW," ",UNPW," ",MYHOST
$ CLOSE GENRPD.DAT
$ OPEN/WRITE FDLIN FIX.FDL
$ WRITE FDLIN "IDENT   ""23-FEB-1988 09:49:43   VAX-11 FDL
Editor"""
$ WRITE FDLIN "    "
$ WRITE FDLIN "SYSTEM"
$ WRITE FDLIN "        SOURCE                    VAX/VMS"
$ WRITE FDLIN "    "
$ WRITE FDLIN "FILE"
$ WRITE FDLIN "        ALLOCATION                391"
$ WRITE FDLIN "        BEST_TRY_CONTIGUOUS       yes"
$ WRITE FDLIN "        EXTENSION                 39"
$ WRITE FDLIN "        ORGANIZATION              sequential"
$ WRITE FDLIN "    "
$ WRITE FDLIN "RECORD"
$ WRITE FDLIN "        BLOCK_SPAN                yes"
$ WRITE FDLIN "        CARRIAGE_CONTROL          carriage_return"
$ WRITE FDLIN "        FORMAT                    fixed"
$ WRITE FDLIN "        SIZE                      80"
$ CLOSE FDLIN
$ CONVERT/PAD=%O40/FDL=FIX GENRPD.DAT GENRPD.DAT
$ DELETE FIX.FDL;0
$!
$ ASSIGN/USER_MODE SYS$COMMAND SYS$INPUT
$ ASSIGN 'LUW'.RPD SYS$OUTPUT
$ RUN CMDIR:[RUNAREA]GENRPD.EXE
$ DEASSIGN SYS$OUTPUT
$ WRITE SYS$OUTPUT "    "
$ WRITE SYS$OUTPUT ">>>GENERATION FINISHED<<<"
$ WRITE SYS$OUTPUT " RESULTS OF GENERATION CAN BE"
$ WRITE SYS$OUTPUT " FOUND ON FILE ''LUW'.RPD"
```

```
$!
$!    MCMPDDL.COM
$!
$!    PERFORM ORACLE V5 PRECOMPILATION ON P1
$!    DEFAULT EXTENSION IS .PC
$!    WHICH PRODUCES P1.C (PRECOMPILED C SOURCE)
$!    AND P1.ERR (PRECOMPILER ERROR LISTING).
$!
$!    PERFORM C COMPILATION ON P1.C
$!    INSERT .OBJ INTO IISSGLIB
$!
$!
$DEFINE IISSGLIB "CDMDIR:[NDDL]GENOLB.OLB"
$IF P1 .EQS. "" THEN INQUIRE P1 "_File "
$IF (F$LOC(".",P1) .EQ. F$LENGTH(P1)) THEN P2=P1+".PC"
$IF (F$LOC(".",P1) .EQ. F$LENGTH(P1)) THEN P3=P1
$IF (F$LOC(".",P1) .NE. F$LENGTH(P1)) THEN P2=P1
$IF (F$LOC(".",P1) .NE. F$LENGTH(P1)) THEN
P3=(F$EXTRACT(0,F$LOC(".",P1),P2))
$INQUIRE CB "DELETE GENERATED .C? (Y/N:  Y IS DEFAULT)
$IF CB .EQS. "" THEN CB="Y"
$PCC INAME='P2' LNAME='P3'.ERR USERID=CDM/CDM INCLUDE=SYS$ORACLE:
HOST=C MAXLITERAL=160 REBIND=YES
$ ON ERROR THEN CONTINUE
$vcc/debug/list='P3'.lis/standard=portable/noopt/DEFINE=VAX 'P3'.C
$ ON ERROR THEN CONTINUE
$LIB/REPLACE IISSGLIB 'P3'.OBJ
$IF CB .EQS. "Y" THEN DELETE 'P3'.C;*
$DELETE  'P3'.LIS;*, 'P3'.ERR;*, 'P3'.OBJ;*
$ WRITE SYS$OUTPUT "  "
$ WRITE SYS$OUTPUT "*** Oracle precompile, C compile, and "
$ WRITE SYS$OUTPUT "    insertion into IISSGLIB of ", P2, "
complete."
$ WRITE SYS$OUTPUT "  "
$ DEFINE IISSGLIB "CDMDIR:[TEST]GENOLB.OLB"
$EXIT
```

```
$!
$!         LNKNDDLC.COM
$!
$!         CREATED APRIL 1988 TO LINK THE NDDL EXECUTABLE USING 'C'
RP-SUBS
$!
$!
$WRITE SYS$OUTPUT " - LINKING NDDL.EXE "
$ ASSIGN NDDL.LINK SYS$OUTPUT
$ IF F$SEARCH("CMDIR:[RUNAREA]NDDL.EXE") THEN GOTO PROCEED
$ DELETE CMDIR:[RUNAREA]NDDL.EXE;*
$!
$PROCEED:
$ DEFINE IISSGLIB CDMDIR:[NDDL]GENOLB.OLB
$!
$STARTLINK:
$@SYS$ORACLE:LFOR CMDIR:[RUNAREA]NDDL -
CDMDIR:[NDDL]NDDL.OBJ,-
CDMDIR:[COM]CDM/OPTIONS,-
SYS$ORACLE:SQLLIB/LIB,-
CDMDIR:[NTMUI]NTMUIOLB.OLB/LIB,-
CDMDIR:[COM]CDMUI.OPT/OPTIONS,-
CDMDIR:[COM]CDMNTM.OPT/OPTIONS SN
$DEASSIGN SYS$OUTPUT
$WRITE SYS$OUTPUT "LINKING COMPLETED"
$ENOUGH:
$DEASSIGN IISSGLIB
$DEASSIGN SYS$OUTPUT
$DELETE NDDL.LINK;*
$DEFINE IISSGLIB "CDMDIR:[TEST]GENOLB.OLB"
$!
```

```
$!
$!         ULNKDDL.COM
$!         THIS USES ORACLE VERSION 5.1
$!
$!   MODIFIED OCTOBER 23,1987-RES FOR RESTRUCTURING
$!
$WRITE SYS$OUTPUT " - LINKING UINDDLZZ.EXE "
$ ASSIGN NDDL.LINK SYS$OUTPUT
$DELETE CMDIR:[RUNAREA]UINDDLZZ.EXE;*
$!
$DEFINE IISSGLIB "CDMDIR:[NDDL]GENOLB.OLB"
$!
$STARTLINK:
$@SYS$ORACLE:LFOR CMDIR:[RUNAREA]UINDDLZZ -
CDMDIR:[NDDL]UINDDL.OBJ,-
CDMDIR:[COM]CDM/OPTIONS,-
SYS$ORACLE:SQLLIB/LIB,-
CDMDIR:[COM]CDMUI.OPT/OPT,-
CDMDIR:[COM]CDMNTM.OPT/OPTIONS Snn
$DEASSIGN SYS$OUTPUT
$WRITE SYS$OUTPUT "LINKING COMPLETED"
$ENOUGH:
$DEASSIGN SYS$OUTPUT
$DELETE NDDL.LINK;*
$DEFINE IISSGLIB "CDMDIR:[TEST]GENOLB.OLB"
$!
```


☆U.S. GOVERNMENT PRINTING OFFICE: 1992 - 648-127/62339